



Guía de Despliegue en Netlify - Radio App

Esta guía te llevará paso a paso para desplegar tu aplicación de radio en Netlify.



Tabla de Contenidos

1. [Prerequisitos](#)
2. [Preparación del Proyecto](#)
3. [Opción 1: Despliegue desde la Web UI \(Recomendado\)](#)
4. [Opción 2: Despliegue desde la CLI](#)
5. [Configuración de Dominio Personalizado](#)
6. [Configuración de Variables de Entorno](#)
7. [Despliegue Continuo \(CI/CD\)](#)
8. [Monitoreo y Optimización](#)
9. [Solución de Problemas](#)



Prerequisitos

Antes de comenzar, asegúrate de tener:

- **Cuenta de GitHub** con el repositorio `radio-app` (<https://github.com/NaktoG/radio-app>)
- **Cuenta de Netlify** (si no la tienes, créala en <https://app.netlify.com/signup>)
- **Código subido a GitHub** (asegúrate de que tu código esté actualizado en la rama `main`)
- **Archivos de configuración creados:**
 - `netlify.toml` ✓ (ya creado)
 - `src/assets/_redirects` ✓ (ya creado)



Preparación del Proyecto

1. Verificar que los archivos de configuración existen

Ejecuta en tu terminal local:

```
cd /home/ubuntu/github_repos/radio-app

# Verifica que netlify.toml existe
ls -la netlify.toml

# Verifica que _redirects existe
ls -la src/assets/_redirects
```

Resultado esperado: Ambos archivos deben existir.

2. Probar la compilación localmente

```
# Compila la aplicación en modo producción
npm run build:prod
```

Resultado esperado:

- La compilación debe terminar sin errores
- Se debe crear el directorio `dist/radio-app/` con los archivos compilados

3. Verificar el contenido compilado

```
# Ver archivos generados
ls -la dist/radio-app/

# Verificar que _redirects se copió
ls -la dist/radio-app/_redirects
```

Resultado esperado: Debes ver archivos como `index.html`, carpeta `assets/`, archivos `.js`, `.css`, y el archivo `_redirects`.

4. Subir cambios a GitHub (si aún no lo has hecho)

```
# Ver estado de Git
git status

# Agregar archivos nuevos
git add netlify.toml src/assets/_redirects

# Hacer commit
git commit -m "feat: Add Netlify deployment configuration"

# Subir a GitHub
git push origin main
```



Opción 1: Despliegue desde la Web UI (Recomendado)


Esta es la forma más sencilla y visual de desplegar en Netlify.

Paso 1: Ingresar a Netlify

1. Abre tu navegador y ve a: <https://app.netlify.com>
2. Inicia sesión con tu cuenta (deberías ver tu equipo “naktog”)

Paso 2: Crear Nuevo Site

1. En el dashboard principal, busca tu equipo “naktog”
2. Haz clic en “Add new site” (botón verde) → “Import an existing project”

 Ruta visual:

Dashboard → Teams (naktog) → Add new site → Import an existing project

Paso 3: Conectar con GitHub

1. Selecciona **“Deploy with GitHub”**
2. Si es la primera vez, Netlify te pedirá que autorices el acceso a tu cuenta de GitHub
3. Haz clic en **“Authorize Netlify”**
4. Puede que GitHub te pida autenticar con tu contraseña o 2FA

Paso 4: Seleccionar el Repositorio

1. En la lista de repositorios, busca **“radio-app”**
2. Si no aparece, haz clic en **“Configure Netlify on GitHub”** para dar acceso a más repositorios
3. Una vez que lo encuentres, haz clic en él

Paso 5: Configurar Build Settings

Netlify debería detectar automáticamente la configuración de tu `netlify.toml`. Verifica que los campos sean:

```
Base directory: (dejar vacío o "/")
Build command: npm run build:prod
Publish directory: dist/radio-app
```

🌟 **Importante:** Si `netlify.toml` existe, Netlify usará esa configuración automáticamente. Los campos arriba son solo referencia.

Paso 6: Variables de Entorno (Opcional)

Si necesitas configurar variables de entorno (por ejemplo, para URLs de API):

1. Haz clic en **“Show advanced”**
2. Haz clic en **“New variable”**
3. Agrega las variables necesarias:

```
Key: API_URL
Value: https://tu-api-backend.com/api
```

Paso 7: Desplegar

1. Haz clic en **“Deploy [tu-repo-name]”** (botón verde grande)
2. Netlify comenzará a:
 - 📦 Clonar tu repositorio
 - 📦 Instalar dependencias (`npm install`)
 - 🔨 Compilar tu aplicación (`npm run build:prod`)
 - 🚀 Desplegar los archivos

Paso 8: Monitorear el Despliegue

1. Serás redirigido a la página de “Deploy log”
2. Podrás ver el progreso en tiempo real:

```
📋 Building
  └─ Cloning repository...
  └─ Installing dependencies...
  └─ Running build command...
  └─ Deploying to CDN...
```

3. Espera a que aparezca:  **"Site is live"** (usualmente toma 2-5 minutos)

Paso 9: Ver tu Aplicación

1. Una vez desplegado, verás un mensaje como:

```
✓ Site is live  
🌐 https://random-name-123456.netlify.app
```

2. Haz clic en el enlace para ver tu aplicación en vivo
3. 🎉 **¡Felicidades!** Tu aplicación está ahora en línea.



Opción 2: Despliegue desde la CLI

Si prefieres la terminal (más rápido para despliegues futuros):

Paso 1: Instalar Netlify CLI

```
# Instalar globalmente  
npm install -g netlify-cli  
  
# Verificar instalación  
netlify --version
```

Paso 2: Autenticarse

```
# Iniciar sesión  
netlify login
```

Esto abrirá tu navegador para que autorices la CLI.

Paso 3: Inicializar el Proyecto

```
# Navegar al proyecto  
cd /home/ubuntu/github_repos/radio-app  
  
# Inicializar Netlify en el proyecto  
netlify init
```

Responde las preguntas:

```
? What would you like to do?
  > Create & configure a new site

? Team:
  > naktog's team

? Site name (optional):
  > radio-app (o deja vacío para uno aleatorio)

? Your build command (hugo build/yarn run build/etc):
  > npm run build:prod

? Directory to deploy (blank for current dir):
  > dist/radio-app

? Netlify functions folder:
  > (presiona Enter para saltar)
```

Paso 4: Desplegar

```
# Compilar y desplegar en producción
netlify deploy --prod
```

O si prefieres hacerlo en dos pasos:

```
# 1. Compilar localmente
npm run build:prod

# 2. Desplegar la carpeta dist
netlify deploy --prod --dir=dist/radio-app
```

Paso 5: Verificar

```
# Ver información del site
netlify status

# Abrir el site en el navegador
netlify open:site
```



Configuración de Dominio Personalizado

Si tienes un dominio propio (como `radioapp.com`):

Opción A: Dominio en Netlify DNS (Recomendado)

1. Ve a tu site en Netlify Dashboard
2. **Site settings** → **Domain management**
3. Haz clic en **"Add custom domain"**
4. Ingresa tu dominio: `radioapp.com`
5. Netlify te mostrará los nameservers:


```
dns1.p01.nsone.net
```

dns2.p01.nsone.net

dns3.p01.nsone.net

dns4.p01.nsone.net

6. Ve al panel de tu proveedor de dominios (GoDaddy, Namecheap, etc.)
7. Cambia los nameservers por los de Netlify
8. **Espera 24-48 horas** para que los DNS se propaguen
9. Netlify configurará **SSL automáticamente** (HTTPS gratis con Let's Encrypt)

Opción B: Dominio con DNS Externo

1. Ve a tu site en Netlify Dashboard
2. **Site settings** → **Domain management**
3. Haz clic en **"Add custom domain"**
4. Ingresa tu dominio: `radioapp.com`
5. Netlify te dirá que agregues un registro DNS:

Para dominio raíz (radioapp.com):

Tipo: A

Host: @

Value: 75.2.60.5

Para www (www.radioapp.com):

Tipo: CNAME

Host: www

Value: [tu-site].netlify.app

1. Agrega estos registros en tu proveedor de DNS
2. Espera a que los DNS se propaguen (puede tomar hasta 24 horas)
3. Netlify configurará SSL automáticamente



Configuración de Variables de Entorno

Si tu aplicación necesita variables de entorno (APIs, tokens, etc.):

Desde la Web UI

1. Ve a tu site en Netlify
2. **Site settings** → **Environment variables** (en el menú lateral)
3. Haz clic en **"Add a variable"** → **"Add a single variable"**
4. Configura cada variable:

Key: `API_URL`

Value: `https://api.tu-backend.com`

Scopes: ✓ All contexts (Production, Deploy Previews, Branch deploys)

1. Haz clic en **"Create variable"**
2. Repite para cada variable necesaria

Desde la CLI

```
# Agregar variable
netlify env:set API_URL "https://api.tu-backend.com"

# Ver variables
netlify env:list

# Eliminar variable
netlify env:unset API_URL
```

Variables Comunes para Radio App

```
# URL del backend (si usas uno)
API_URL=https://api.tu-dominio.com

# Clave de API de Radio Browser (si es necesario)
RADIO_BROWSER_API_KEY=tu-clave-aqui

# Entorno
NODE_ENV=production
```



Despliegue Continuo (CI/CD)

Una vez conectado con GitHub, Netlify desplegará automáticamente:

Configuración por Defecto

- **Rama main** : Despliega a **producción** (tu site principal)
- **Pull Requests**: Crea **deploy previews** (URLs temporales para revisar cambios)
- **Otras ramas**: Crea **branch deploys** (URLs por rama)

Personalizar Configuración

1. Ve a **Site settings** → **Build & deploy** → **Continuous Deployment**
2. Configura:

Branch to deploy:

Production branch: main

Deploy contexts:

- ✓ Deploy Pull Requests
- ✓ Deploy branch deploys

Build hooks (para triggers externos):

Haz clic en "Add build hook"

Nombre: Deploy from external service

Branch: main





Ignorar Commits

Si quieres que un commit NO despliegue, agrega al mensaje:

```
git commit -m "docs: Update README [skip netlify]"
```

Monitoreo y Optimización

Ver Analytics

1. Ve a tu site en Netlify
2. **Analytics** (en el menú superior)
3. Aquí verás:
 -  Visitantes únicos
 -  Page views
 -  Países de origen
 -  Dispositivos

Optimizar Rendimiento

1. Habilitar Asset Optimization (Gratis)

1. **Site settings** → **Build & deploy** → **Post processing**

2. Activa:

- ✓ Bundle CSS
- ✓ Minify CSS
- ✓ Minify JS
- ✓ Compress images
- ✓ Pretty URLs

2. Configurar Headers de Cache

Ya están configurados en tu `netlify.toml` :

```
# Assets estáticos con cache de 1 año
[[headers]]
  for = "/assets/*"
  [headers.values]
    Cache-Control = "public, max-age=31536000, immutable"

# index.html sin cache
[[headers]]
  for = "/index.html"
  [headers.values]
    Cache-Control = "public, max-age=0, must-revalidate"
```

3. Habilitar Brotli Compression

Netlify lo hace automáticamente, pero puedes verificar:

```
curl -H "Accept-Encoding: br" -I https://tu-site.netlify.app
```

Debes ver: `content-encoding: br`

Lighthouse Score

1. Ve a **Site settings** → **Build & deploy** → **Post processing**

2. Haz clic en **“Get Lighthouse score”**
3. Netlify ejecutará un test de performance y te mostrará:
 - ⚡ Performance
 - ♿ Accessibility
 - 🔍 Best Practices
 - 📱 SEO

! Solución de Problemas

Problema 1: Build Failed - “Command not found: npm”

Error:

```
Error: npm: command not found
```

Solución:

Asegúrate de que `netlify.toml` tenga:

```
[build.environment]
  NODE_VERSION = "18"
```

O configura en **Site settings** → **Build & deploy** → **Environment** → **Environment variables**:

```
NODE_VERSION = 18
```

Problema 2: 404 al Recargar Página

Error: Al navegar a `/stations` funciona, pero al recargar la página da 404.

Causa: Falta configuración de redirects para SPA.

Solución:

Verifica que exista `netlify.toml` con:

```
[[redirects]]
  from = "/*"
  to = "/index.html"
  status = 200
```

O que exista `dist/radio-app/_redirects` con:

```
/*      /index.html  200
```

Verificar:

```
# Ver si _redirects se copió al dist
npm run build:prod
ls -la dist/radio-app/_redirects
```

Problema 3: Build Failed - “Angular CLI not found”

Error:

```
Error: Cannot find module '@angular/cli'
```

Solución:

```
# Agregar @angular/cli a devDependencies
npm install --save-dev @angular/cli

# Commit y push
git add package.json package-lock.json
git commit -m "fix: Add @angular/cli to devDependencies"
git push
```

Problema 4: Build Timeout (15 minutos)

Error:

```
Error: Build exceeded maximum allowed runtime
```

Causas posibles:

- `npm install` muy lento
- Dependencias muy pesadas

Solución:

1. Usar `npm ci` en lugar de `npm install`:

```
toml
[build]
command = "npm ci && npm run build:prod"
```

2. Limpiar cache de `node_modules`:

```
bash
# En Netlify UI
Site settings → Build & deploy → Clear cache and retry deploy
```

Problema 5: Estilos No Cargan (Tailwind)

Error: La app se ve sin estilos.

Causa: Tailwind CSS no se compiló correctamente.

Solución:

Verifica `angular.json` :

```
"styles": [
  "src/styles.css" // ← Debe incluir Tailwind
]
```

Y que `src/styles.css` tenga:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Problema 6: Mixed Content Errors (HTTP/HTTPS)**Error:**

Mixed Content: The page at '`https://...`' was loaded over HTTPS, but requested an insecure resource '`http://...`'

Causa: Tu app intenta cargar recursos HTTP desde una página HTTPS.

Solución:

1. Asegúrate de que todas las URLs en tu código usen HTTPS
2. Para APIs externas, usa siempre HTTPS
3. Para Radio Browser API, usa:

```
typescript
```

```
apiUrl: 'https://del.api.radio-browser.info' // ✅ HTTPS
```

Problema 7: Build Exitoso pero App en Blanco

Error: Build termina OK, pero al abrir el site solo ves una página blanca.

Diagnóstico:

1. Abre DevTools (F12) → Console
2. Busca errores JavaScript

Soluciones comunes:**a) Error de rutas base:**

```
typescript
```

```
// En angular.json
```

```
"baseHref": "/" // Asegúrate de que sea "/"
```

b) Error de producción:

```
bash
```

```
# Probar localmente con producción
```

```
npm run build:prod
```

```
cd dist/radio-app
npx http-server -p 8080
```

c) Error de TypeScript:

```
bash
# Ver errores de compilación
ng build --configuration production --verbose
```

Problema 8: Deploy Preview No Aparece en PR

Error: Abres un Pull Request pero no ves el link de deploy preview.

Solución:

1. Ve a **Site settings** → **Build & deploy** → **Deploy contexts**

2. Asegúrate de que esté activado:

✓ Deploy Pull Requests

3. Verifica permisos de GitHub:

- Ve a <https://github.com/settings/installations>
- Busca "Netlify"
- Asegúrate de que tenga acceso al repositorio

Problema 9: Error de CORS en Producción

Error:

```
Access to fetch at 'https://api.backend.com' from origin 'https://tu-site.netlify.app'
has been blocked by CORS policy
```

Causa: Tu backend no permite requests desde tu dominio de Netlify.

Solución:

Opción A: Configurar CORS en tu backend (recomendado)

```
// Ejemplo en Node.js/Express
app.use(cors({
  origin: ['https://tu-site.netlify.app', 'https://tu-dominio.com'],
  credentials: true
}));
```

Opción B: Usar Netlify como Proxy

En `netlify.toml`:

```
[[redirects]]
  from = "/api/*"
  to = "https://tu-backend.com/api/:splat"
  status = 200
  force = true
```

Luego en tu código Angular:

```
// Antes
apiUrl: 'https://tu-backend.com/api'

// Después (usa ruta relativa)
apiUrl: '/api'
```

Problema 10: Funciona en Dev pero No en Producción

Error: `ng serve` funciona perfectamente, pero en Netlify no.

Pasos de diagnóstico:

1. Probar build de producción localmente:

```
bash
npm run build:prod
cd dist/radio-app
npx http-server -p 8080
```

2. Ver logs de Netlify:

- Ve a tu site → **Deploys** → Click en el último deploy
- Revisa la sección "Deploy log"
- Busca warnings o errors

3. Habilitar modo de debug:

```
toml
[build]
command = "npm run build:prod -- --verbose"
```

4. Verificar variables de entorno:

- Dev usa `environment.ts`
- Prod usa `environment.prod.ts`
- Asegúrate de que ambos estén correctos

Checklist Final

Antes de desplegar, verifica:

- ☐ ☒ `netlify.toml` existe en la raíz
- ☐ ☒ `src/assets/_redirects` existe
- ☐ ☒ `npm run build:prod` compila sin errores
- ☐ ☒ `dist/radio-app/_redirects` se genera después del build
- ☐ ☒ Código subido a GitHub
- ☐ ☒ Cuenta de Netlify creada
- ☐ ☒ Site conectado a GitHub
- ☐ ☒ Build settings configurados correctamente
- ☐ ☒ Variables de entorno configuradas (si aplica)

- [] ☒ SSL configurado (automático en Netlify)
- [] ☒ Dominio personalizado configurado (opcional)

¡Listo para Desplegar!

Si seguiste todos los pasos, tu aplicación debería estar funcionando en:

 `https://[tu-site-name].netlify.app`

URLs Importantes

- **Dashboard de Netlify:** <https://app.netlify.com/teams/naktog/sites>
- **Tu Site:** [https://app.netlify.com/sites/\[tu-site-name\]](https://app.netlify.com/sites/[tu-site-name])
- **Repositorio GitHub:** <https://github.com/NaktoG/radio-app>
- **Logs de Deploy:** [https://app.netlify.com/sites/\[tu-site-name\]/deploys](https://app.netlify.com/sites/[tu-site-name]/deploys)

Próximos Pasos

1. **Prueba tu aplicación** en todos los navegadores
2. **Configura un dominio personalizado** (si lo tienes)
3. **Activa Analytics** para ver estadísticas
4. **Optimiza el rendimiento** con Lighthouse
5. **Configura notificaciones** de deploy (Slack, Email)

Recursos Adicionales

- [Documentación oficial de Netlify](https://docs.netlify.com/) (<https://docs.netlify.com/>)
- [Netlify CLI Reference](https://cli.netlify.com/) (<https://cli.netlify.com/>)
- [Angular Deployment Guide](https://angular.io/guide/deployment) (<https://angular.io/guide/deployment>)
- [Netlify Community Forums](https://answers.netlify.com/) (<https://answers.netlify.com/>)
- [Netlify Status](https://www.netlifystatus.com/) (<https://www.netlifystatus.com/>)

¿Necesitas Ayuda?

Si encuentras algún problema que no está en esta guía:

1. **Revisa los logs de Netlify:** Casi siempre tienen la respuesta
2. **Busca en Netlify Forums:** <https://answers.netlify.com/>
3. **Revisa el GitHub Issues:** <https://github.com/NaktoG/radio-app/issues>
4. **Contacta soporte de Netlify:** <https://www.netlify.com/support/>

¡Feliz despliegue! 

Última actualización: 22 de octubre de 2025

Versión del proyecto: 2.0.0

Angular: 16.2.0

Netlify CLI: Latest